# METHOD AND APPARATUS FOR DATA ANALYSIS

## BACKGROUND OF THE INVENTION

Data analysis is used in many different areas, such as data mining, statistical analysis, artificial intelligence, machine learning, and process control to provide information that can be applied to different environments. Usually this analysis is performed on a collection of data organised in a database. With large databases, computations required for the analysis often take a long time to complete.

Databases can be used to determine relationships between variables and provide a model that can be used in the data analysis. These relationships allow the value of one variable to be predicted in terms of the other variables. Minimizing computational time is not the only requirement for successful data analysis. Overcoming rapid obsolescence of models is another major challenge.

Currently tasks such as prediction of new conditions, process control, fault diagnosis and yield optimization are done using computers or microprocessors directed by mathematical models. These models generally need to be "retrained" or "recalibrated" frequently in dynamic environments because changing environmental conditions render them obsolete. This situation is especially serious when very large quantities of data are involved or when large changes to the models are required over short periods of time. Obsolescence can originate from new data values being drastically different from historical data because of an unforeseen change in the environment of a sensor, one or more sensors becoming inoperable during operation or new sensors being added to a system for example.

In real-world applications, there are several other requirements that often become vital in addition to computational speed and rapid model obsolescence. For example, in some cases the model will need to deal with a stream of data rather than a static database. Also, when databases are used they can rapidly outgrow the available computer storage available. Furthermore, existing computer facilities can become insufficient to accomplish model re-calibration. Often it becomes completely impractical to use a whole database for re-calibration of the model. At some risk, a sample is taken from the database and used to obtain the re-calibrated model. In developing models, "scenario testing" is often used. That is, a variety of models need to be tried on the data. Even with moderately sized databases this can be a processing intensive task. For example, although combining variables in a model to form a new model is very attractive from an efficiency viewpoint (termed here "dimension reduction"), the number of possible combinations combined with the data processing usually required for even one model, especially with a large database, makes the idea impractical with current methods. Finally, often models are used in situations where they must provide an answer very quickly, sometimes with inadequate data. In credit scoring for example, a large number of risk factors can affect the credit rating and the interviewer wishes to obtain the answer from a credit assessment model as rapidly as possible with a minimum of data. Also, in medical diagnosis, a doctor would like to converge on the solution with a minimum of questions. Methods which can request the data needed based on maximizing the probability of arriving at a conclusion as quickly as possible (termed here "dynamic query") would be very useful in many diagnostic applications.

Finally, mobile applications are now becoming very important in technology. A method of condensing the knowledge in a large database so that it can be used with a model in a portable device is highly desirable.

This situation is becoming increasingly important in an extremely diverse range of areas ranging from finances to health care and from sports forecasting to retail needs.

## FIELD OF THE INVENTION

The present invention relates to a method and apparatus for data analysis.

## DESCRIPTION OF THE PRIOR ART

The primary focus in the previous art has been to focus upon reducing computational time. Recent developments in database technology are beginning to emphasize "automatic summary tables" ("AST's") that contain pre-computed quantities needed by "queries" to the database. These AST's provide a "materialized view" of the data and greatly increase the speed of response to queries. Efficiently updating the AST's with new data records, as the new data becomes available for the database has been the subject of many publications. Initially only very simple queries were considered. Most recently incrementally updating an AST in accordance with a method of updating AST's that applies to all "aggregate functions" has been proposed. However, although the AST's speed up the response to queries, they are still very extensive compilations of data and therefore incremental re-computation is generally a necessity for their maintenance. Palpanas et al. proposed what they term as "the first" general algorithm to efficiently re-compute only the groups in the AST which need to be updated in order to reply to the query. However, their method is a very involved one. It includes a considerable amount of work to select the groups that are to be updated. Their experiments indicate that their method runs in 20% to 60% of the time required for a "full refresh" of the AST. There is increasing interest in using AST's to respond to queries that originate from On-line Analytical Processing ("OLAP"). These can involve standard statistical or data-mining methods.

Chen et al. examined the problem of applying OLAP to dynamic rather than static situations. In particular, they were interested in multi-dimensional regression analysis of time-series data streams. They recognized that it should be possible to use only a small number of pre-computed quantities rather than all of the data. However, the algorithms that they propose are very involved and constrained in their utility.

U.S. Patent 6,553,366 shows how great economies of data storage requirements and time can be obtained by storing and using various "scalable data mining functions" computed from a relational database. This is the most recent version of the "automatic summary table" idea.

Thus, although the prior art has recognized that pre-computing quantities needed in subsequent modeling calculations saves time and data storage, the methods developed fail to satisfy some or all of the other requirements mentioned above. Often they can add records but cannot remove records to their "static" databases. Adding new variables or removing variables "on the fly" (in real time) is not generally known. They are not used to combine databases or for parallel processing. Scenario testing is very limited and does not involve dimension reduction. Dynamic query is not done with static decision trees being commonplace. Methods are generally embedded in large office information systems with so many quantities computed and so many ties to existing interfaces that portability is challenging.

It is therefore an object of the present invention to provide a method of and apparatus for data analysis that obviates or mitigates some of the above disadvantages.

## SUMMARY OF THE INVENTION

In one aspect, the present invention provides a "knowledge entity" that may be used to perform incremental learning. The knowledge entity is conveniently represented as a matrix where one dimension represents independent variables and the other dimension represents dependent variables. For each possible pairing of variables, the knowledge entity stores selected combinations of either or both of the variables. These selected combinations are termed the "knowledge elements" of the knowledge entity. This knowledge entity may be updated efficiently with new records by matrix addition. Furthermore, data can be removed from the knowledge entity by matrix subtraction. Variables can be added or removed from the knowledge entity by adding or removing a set of cells, such as a row or column to one or both dimensions.

Preferably the number of joint occurrences of the variables is stored with the selected combinations.

Exemplary combinations of the variables are the sum of values of the first variable for each joint occurrence, the sum of values of the second variable for each joint occurrence, and the sum of the product of the values of each variable.

In one further aspect of the present invention, there is provided a method of performing a data analysis by collecting data in such the knowledge entity and utilising it in a subsequent analysis.

According to another aspect of the present invention, there is provided a process modelling system utilising such the knowledge entity.

According to other aspects of the present invention, there is a provided either a learner or predictor using such the knowledge entity.

The term "analytical engine" is used to describe the knowledge entity together with the methods required to use it to accomplish incremental learning operations, parallel processing operations, scenario testing operations, dimension reduction operations, dynamic query operations and/or distributed processing operations. These methods include but are not limited to methods for data collecting, management of the knowledge elements, modelling and use of the modelling (for prediction for example). Some aspects of the management of the knowledge elements may be delegated to a conventional data management system (simple summations of historical data for example). However, the knowledge entity is a collection of knowledge elements specifically selected so as to enable the knowledge entity to accomplish the desired operations. When modeling is accomplished using the knowledge entity it is referred to as "intelligent modeling" because the resulting model receives one or more characteristics of intelligence. These characteristics include: the ability to immediately utilize new data, to purposefully ignore some data, to incorporate new variables, to not use specific variables and, if necessary, to do be able to utilize these characteristics on-line (at the point of use) and in real time.

## BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described by way of example only with reference to the accompanying drawings in which:

Figure 1 is a schematic diagram of a processing apparatus;

Figure 2 is a representation of a controller for the processing apparatus of Figure 1;

Figure 3 is a schematic of a the knowledge entity used in the controller of Figure 2;

Figure 4 is a flow chart of a method performed by the controller of Figure 2;

Figure 5 is another flow chart of a method performed by the controller of Figure 2;

Figure 6 is a further flow chart of a method performed by the controller of Figure 2;

Figure 7 is a yet further flow chart of a method performed by the controller of Figure 2;

Figure 8 is a still further flow chart of a method performed by the controller of Figure 2;

Figure 9 is a schematic diagram of a robotic arm;

Figure 10 is a schematic diagram of a Markov chain;

Figure 11 is a schematic diagram of a Hidden Markov model;

Figure 12 is another schematic diagram of a Hidden Markov model.

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

To assist in understanding the concepts embodied in the present invention and to demonstrate the industrial applicability thereof with its inherent technical effect, a first embodiment will describe how the analytical engine enables application to the knowledge entity of incremental learning operations for the purpose of process monitoring and control. It will be appreciated that the form of the processing apparatus is purely for exemplary purposes to assist in the explanation of the use of the knowledge entity shown in Figure 3, and is not intended to limit the application to the particular apparatus or to process control environments. Subsequent embodiments will likewise illustrate the flexibility and general applicability in other environments.

Referring therefore to Figure 1, a dryer 10 has a feed tube 12 for receiving wet feed 34. The feed tube 12 empties into a main chamber 30. The main chamber 30 has a lower plate 14 to form a plenum 32. An air inlet 18 forces air into a heater 16 to provide hot air to the plenum 32. An outlet tube 28 receives dried material from the main chamber 30. An air outlet 20 exhausts air from the main chamber 32.

The dryer 10 is operated to produce dried material, and it is desirable to control the rate of production. An exemplary operational goal is to produce 100 kg of dried material per hour.

The dryer receives wet feed 34 through the feed tube 12 at an adjustable and observable rate. The flow rate from outlet tube 28 can also be monitored. The flow rate from outlet tube 28 is related to operational parameters such as the wet feed flow rate, the temperature provided by heater 16, and the rate of air flow from air inlet 18. The dryer 10 incorporates a sensor for each operational parameter, with each sensor connected to a controller 40 shown in detail in Figure 2. The controller 40 has a data collection unit 42, which receives inputs from the sensors associated with the wet feed tube 12, the heater 16, the air inlet 18, and the output tube 28 to collect data.

The controller 40 has a learner 44 that processes the collected data into a knowledge entity 46. The knowledge entity 46 organises the data obtained from the operational parameters and the output flow rate. The knowledge entity 46 is initialised to notionally contain all zeroes before its first use. The controller 40 uses a modeller 48 to form a model of the collected data from the knowledge entity 46. The controller 40 has a predictor 50 that can set the operational parameters to try to achieve the operational goal. Thus, as the controller operates the dryer 10, it can monitor the production and incrementally learn a better model.

The controller 40 operates to adjust the operational parameters to control the rate of production. Initially the dryer 10 is operated with manually set operational parameters. The initial operation will produce training data from the various sensors, including output rate.

The data collector 42 receives signals related to each of the operational parameters and the output rate, namely a measure of the wet feed rate from the wet feed tube 12, a measure of the air temperature from the heater 16, a measure of the air flow from the air inlet 18, and a measure of the output flow rate from the output tube 28.

The learner 44 transforms the collected data into the knowledge entity of Figure 3 as each measurement is received. As can be seen in Figure 3, the knowledge entity 46 is organised as an orthogonal matrix having a row and a column for each of the sensed operating parameters. The intersection of each row and column defines a cell in which a set of combinations of the variable in the respective row and column is accumulated.

In the embodiment of Figure 3, for each pairing of variables, a set of four combinations is obtained. The first combination, $n_{i,j}$ is a count of the number of joint occurrences of the two variables. The combination $\sum X_i$ represents the total of all measurements of the first variable $X_i$, which is one of the sensed operational parameters. The second quantity $\sum X_j$ records the total of all measurements of the second variable $X_j$, which is another of the sensed operational parameters. Finally, $\sum X_i X_j$ records the total of the products of all measurements of both variables. It is noted that the summations are over all observed measurements of the variables.

These combinations are additive, and accordingly can be computed incrementally. For example, given observed measurements [3, 4, 5, 6] for the variable

$X_i$, then $\sum X_i = 3+4+5+6=18$. If the measurements are subdivided into two collections of observed measurements [3, 4] and [5, 6], for example from sensors at two different locations, then $\sum_{[3,4]} X_i = 7$ and $\sum_{[5,6]} X_i = 11$ so $\sum_{[3,4,5,6]} X_i = \sum_{[3,4]} X_i + \sum_{[5,6]} X_i$. The nature of the subdivision is not relevant, so the combination can be computed incrementally for successive measurements, and two collections of measurements can be combined by addition of their respective combinations.

In general, the combinations of parameters accumulated should have the property that given a first and second collection of data, the value of the combination of the collections may be efficiently computed from the values of the collections themselves. In other words, the value obtained for a combination of two collections of data may be obtained from operations on the value of the collections rather than on the individual elements of the collections.

It is also recognised that the above combinations have the property that given a collection of data and additional data, which can be combined into an augmented collection of data, the value of the combination for the augmented collection of data is efficiently computable from the value of the combination for the collection of data and the value of the combination for the additional data. This property allows combination of two collections of measurements.

An example of data received by the data collector 42 from the dryer of Figure 1 in four separate measurements is as follows:

| Measurement | Wet Feed Rate | Air Temperature | Air Flow | Dry Output Rate |
|---|---|---|---|---|
| 1 | 10 | 30 | 110 | 2 |
| 2 | 15 | 35 | 115 | 3 |
| 3 | 5 | 40 | 120 | 1.5 |
| 4 | 15 | 50 | 140 | 6 |

Table 1

With the measurements shown above in Table 1, measurement 1 is transformed into the following record represented as an orthogonal matrix:

| Measurement 1 | Wet Feed Rate | Air Temperature | Air Flow | Dry Output Rate |
|---|---|---|---|---|
| Wet Feed Rate | $1 = n_{11}$<br>$10 = x_1$<br>$10 = x_2$<br>$100 = x_1x_2$ | 1<br>10<br>30<br>300 | 1<br>10<br>110<br>1100 | 1<br>10<br>2<br>20 |
| Air Temperature | 1<br>30<br>10<br>300 | 1<br>30<br>30<br>900 | 1<br>30<br>110<br>3300 | 1<br>30<br>2<br>60 |
| Air Flow | 1<br>110<br>10<br>1100 | 11<br>110<br>30<br>3300 | 1<br>110<br>110<br>12100 | 1<br>110<br>2<br>220 |
| Dry Output Rate | 1<br>2<br>10<br>20 | 1<br>2<br>30<br>60 | 1<br>2<br>110<br>220 | 1<br>2<br>2<br>4 |

Table 2

This measurement is added to the knowledge entity 46 by the learner 42. Each subsequent measurement is transformed into a similar table and added to the knowledge entity 46 by the learner 42.

For example, upon receipt of the second measurement, the cell at the intersection of the wet feed row and air temperature column would be updated to contain:

| | Air Temperature |
|---|---|
| Wet Feed Rate | 1+1=2<br>10 + 15 =25<br>30 + 35 = 65<br>300 + 525 = 825 |

Table 3

Successive measurements can be added incrementally to the knowledge entity 46 since the knowledge entity for a new set of data is equal to the sum of the knowledge entity for an old set data with the knowledge entity of the additional data. Each of the combinations F used in the knowledge entity 46 have the exemplary property that $F(A \cup B) = F(A) + F(B)$ for sets A and B. Further properties of the knowledge entity 46 will be discussed in more detail below.

As data are collected, the controller 40 accumulates data in the knowledge entity 46 which may be used for modelling and prediction. The modeller 48 determines the parameters of a predetermined model based on the knowledge entity 46. The predictor 50 can then use the model parameters to determine desirable settings for the operational parameters.

After the controller 40 has been trained, it can begin to control the dryer 10 using the predictor 50. Suppose that the operator instructs the controller 40 through the user interface 52 to set the production rate to 100 kg/h by varying the air temperature at heater 16, and that the appropriate control method uses a linear regression model.

The modeller 48 computes regression coefficients as shown in Figure 4 generally by the numeral 100. At step 102, the modeller computes a covariance table. Covariance between two variables $X_i$ and $X_j$ may be computed as

$Covar_{i,j} = \frac{\sum X_i X_j - \frac{\sum X_i \sum X_j}{n_{ij}}}{n_{ij}}$. Since each of these terms is one of the combinations stored in the knowledge entity 46 at the intersection of row i and column j, computation of the covariance for each pair of variables is done with two divisions and one subtraction. When $i = j$, the covariance is equal to the variance, i.e. $Covar_{i,j} = Var_i = Var_j$. The modeller 48 uses this relationship to compute the covariance between each pair of variables.

Then at step 104, the modeller 48 computes a correlation table. The correlation between two variables $X_i$ and $X_j$ may be computed as $R_{i,j} = \frac{Covar_{i,j}}{\sqrt{Var_i Var_j}}$. Since each of these terms appears in the covariance table obtained from the knowledge entity 46 at step 102, the correlation coefficient can be computed with one multiplication, one square root, and one division. The modeller 48 uses this relationship to compute the correlation between each pair of variables.

At step 106, the operator selects a variable Y, for example $X_4$, to model through the user interface 52. At step 107, the modeller 48 computes $\beta = R_{i,j}^{-1} R_{y,j}$ using the entries in the correlation table.

At step 108, the modeller 48 first computes the standard deviation $s_y$ of the dependent variable Y and the standard deviation $s_j$ of independent variables $X_j$. Conveniently, the standard deviations $s_y = \sqrt{Var_y}$ and $s_j = \sqrt{Var_j}$ are computed using the entries from the covariance  table. The modeller 48 then computes the coefficients $b_j = \beta_j \left( \frac{s_y}{s_j} \right)$.

At step 109, the modeller 48 computes an intercept $a = \overline{X_4} - b_1 \overline{X_1} - b_2 \overline{X_2} - b_3 \overline{X_3}$. The modeller 48 then provides the coefficients a, $b_1$, $b_2$, $b_3$ to the predictor 50.

The predictor 50 can then estimate the dependent variable as
$Y = a + b_1\overline{X_1} + b_2\overline{X_2} + b_3\overline{X_3}$.

The knowledge entity shown in Figure 3 provides the analytical engine significant flexibility in handling varying collections of data. Referring to Figure 5 a method of amalgamating knowledge from another controller is shown generally by the numeral 110. The controller 40 first receives at step 112 a new knowledge entity from another controller. The new knowledge entity is organised to be of the same form as the existing knowledge entity 46. This new knowledge entity may be based upon a similar process in another factory, or another controller in the same factory, or even standard test data or historical data. The controller 40 provides at step 114 the new knowledge entity to learner 44. Learner 44 adds the new knowledge to the knowledge entity 46 at step 116. The new knowledge is added by performing a matrix addition (i.e. addition of similar terms) between the knowledge entity 46 and the new knowledge entity. Once the knowledge entity 46 has been updated, the model is updated at step 118 by the modeller 48 based on the updated knowledge entity 46

In some situations it may be necessary to reverse the effects of amalgamating knowledge shown in Figure 5. In this case, the method of Figure 6 may be used to remove knowledge. Referring therefore to Figure 6, a method of removing knowledge from the knowledge entity 46 is shown generally by the numeral 120. To begin, at step 122, the controller 40 accesses a stored auxiliary knowledge entity. This may be a record of previously added knowledge from the method of Figure 5. Alternatively, this may be a record of the knowledge entity at a specific time. For example, it may be desirable to eliminate the knowledge added during the first hour of operations, as it may relate to startup conditions in the plant which are considered irrelevant to future modelling. The stored auxiliary knowledge entity has the same form as the knowledge entity 46 shown in Figure 3. The controller 40 provides the auxiliary knowledge entity to the learner 44 at step 124. The learner 44 at step 126 then removes the auxiliary knowledge from the knowledge entity 46 by subtracting the auxiliary knowledge entity from knowledge entity 46. Finally at step 128, the model is updated with the modified knowledge entity 46.

To further refine the modelling, an additional sensor may be added to the dryer 10. For example, a sensor to detect humidity in the air inlet may be used to consider the effects of external humidity on the system. In this case, the model may be updated by performing the method shown generally by the numeral 130 in Figure 7. First a new sensor is added at step 132. The learner 44 then expands the knowledge entity by adding a row and a column. The combinations in the new row and the new column have notional values of zero. The controller 44 then proceeds to collect data at step 136. The collected data will include that obtained from the old sensors and that of the new sensor. This information is learned at step 138 in the same manner as before. The knowledge entity 46 in the analytical engine can then be used with the new sensor to obtain the coefficients of the linear regression using all the sensors including the new sensor. It will be appreciated that since the values of 'n' in the new row and column initially are zero, that there will be a significant difference between the values of 'n' in the new row and column and in the old rows and columns. This difference reflects that more data has been collected for the original rows and columns. It will therefore be recognised that provision of the value of 'n' contributes to the flexibility of the knowledge entity.

It may also be desirable to eliminate a sensor from the model. For example, it may be discovered that air flow does not affect the output speed, or that air flow may be too expensive to measure. The method shown generally as 140 in Figure 7 allows an operational parameter to be removed from the knowledge entity 46. At step 142, an operational parameter is no longer relevant. The operational parameter corresponds to a variable in the knowledge entity 46. The learner 44 then contracts the knowledge entity at step 144 by deleting the row and column corresponding to the removed variable. The model is then updated at step 146 to obtain the linear regression coefficients for the remaining variable to eliminate use of the deleted variable.

It will be noted in each of these examples that the updates is accomplished without requiring a summing operation for individual values of each of the previous records. Similarly subtraction is performed without requiring a new summing operation for the remaining records. . No substantial re-training or re-calibration is required.

## DISTRIBUTED AND PARALLEL DATA PROCESSING

A particularly useful attribute of the knowledge entity 46 in the analytical engine is that it allows databases to be divided up into groups of records with each group processed separately, possibly in separate computers. After processing, the results from each of these computers may be combined to achieve the same result as though the whole data set had been processed all at once in one computer. The analytical engine is constructed so as to enable application to the knowledge entity of such parallel processing operations. This can achieve great economies of hardware and time resources. Furthermore, instead of being all from the one database, some of these groups of records can originate from other databases. That is, they may be "distributed" databases. The combination of diverse databases to form a single knowledge entity and hence models which draw upon all of these databases is then enabled. That is, the analytical engine enables application to the knowledge entity of distributed processing as well as parallel processing operations.

As an illustration, if the large database (or distributed databases) can be divided into ten parts then these parts may be processed on computers 1 to 10 inclusive, for example. In this case, these computers each process the data and construct a separate knowledge entity. The processing time on each of these computers depends on the number of records in each subset but the time required by an eleventh computer to combine the records by processing the knowledge entity is small (usually a few milliseconds). For example, with a dataset with 1 billion records that normally requires 10 hours to process in a single computer, the processing time can be decreased to 1 hour and a few seconds by subdividing the dataset into ten parts.

To demonstrate this attribute, the following example considers a very small dataset of six records and an example of interpretation of dryer output rate data from three dryers. If, for example, the output rate from the third dryer is to be predicted from the output rate from the other two dryers then an equation is required relating it to these other two output rates. The data is shown in the table below where $X_1$, $X_2$ and $X_3$ represent the three output rates. The sample dataset with six records and three variables is set forth below at Table 4.

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| 2 | 3 | 5 |
| 3 | 4 | 7 |
| 1 | 1 | 3 |
| 2 | 3 | 6 |
| 4 | 4 | 8 |
| 3 | 5 | 7 |

Table 4

With such a small amount of data it is practical to use multiple linear regression to obtain the needed relationship:

Multiple linear regression for the dataset shown in Table 4 provides the relationship:

$$X_3 = 1.652 + 1.174 * X_1 + 0.424 * X_2$$

However, if this dataset consisted of a billion records instead of only six then multiple linear regression on the whole dataset at once would not be practical. The conventional approach would be to take only a random sample of the data and obtain a multiple linear regression model from that, hoping that the resulting model would represent the entire dataset.

Using the knowledge entity 46, the analytical engine can use the entire dataset for the regression model, regardless of the size of the data set. This can be illustrated using only the six records shown as follows and dividing the dataset into only three groups.

***Step 1:*** Divide the dataset to three subsets with two records in each, and compute a knowledge entity for each subset. The data in subset 1 has the form shown below in Table 5.

**Subset 1:**

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| 2 | 3 | 5 |
| 3 | 4 | 7 |

Table 5

From the data in Table 5 above, a knowledge entity I (Table 6) is calculated for subset 1

(Table 5) using a first computer.

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $X_1$ | 2<br>5<br>5<br>13 | 2<br>5<br>7<br>18 | 2<br>5<br>12<br>31 |
| $X_2$ | 2<br>7<br>5<br>18 | 2<br>7<br>7<br>25 | 2<br>7<br>12<br>43 |
| $X_3$ | 2<br>12 | 2<br>12 | 2<br>12 |

| | | | |
|---|---|---|---|
| | 5 | 7 | 12 |
| | 31 | 43 | 74 |

Table 6

As described above, the knowledge entity 46 is built by using the basic units which includes an input variable $X_j$ an output variable $X_i$ and a set of combinations indicated as $W_{ij}$, as shown in Table 7:

| | $X_j$ |
|---|---|
| $X_i$ | $W_{ij}$ |

Table 7

Where $W_{ij}$ includes one or more of the following four basic elements:

$N_{ij}$ is the total number of joint occurrence of two variables

□ $X_i$ is the sum of variable $X_i$

□ $X_j$ is the sum of variable $X_j$

□ $X_i X_j$ is the sum of multiplication of variable $X_i$ and $X_j$

In some applications it may be advantageous to include additional knowledge elements for specific calculation reasons. For example: □ $X^3$, □ $X^4$ and□ $(X_i X_j)^2$ can generally be included in the knowledge entity in addition to the four basic elements mentioned above without adversely affecting the intelligent modeling capabilities.

The data in subset 2 has the form shown below in Table 8.

**Subset 2:**

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| 1 | 1 | 3 |

| | | |
|---|---|---|
| 2 | 3 | 6 |

Table 8

A knowledge entity II (Table 9) is calculated for subset 2 (Table 8) using a second computer.

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $X_1$ | 2<br>3<br>3<br>5 | 2<br>3<br>4<br>7 | 2<br>3<br>9<br>15 |
| $X_2$ | 2<br>4<br>3<br>7 | 2<br>4<br>4<br>10 | 2<br>4<br>9<br>21 |
| $X_3$ | 2<br>9<br>3<br>15 | 2<br>9<br>4<br>21 | 2<br>9<br>9<br>45 |

Table 9

Similarly, for subset 3 shown in Table 10, a knowledge entity III (Table 11) is computed using a third computer.

**Subset 3:**

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| 4 | 4 | 8 |
| 3 | 5 | 7 |

Table 10

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $X_1$ | 2<br>7<br>7<br>25 | 2<br>7<br>9<br>31 | 2<br>7<br>15<br>53 |
| $X_2$ | 2<br>9<br>7<br>31 | 2<br>9<br>9<br>41 | 2<br>9<br>15<br>67 |
| $X_3$ | 2<br>15<br>7<br>53 | 2<br>15<br>9<br>67 | 2<br>15<br>15<br>113 |

Table 11

***Step 2:*** Calculate a knowledge entity IV (Table 12) by adding together the three previously calculated knowledge tables using a fourth computer.

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $X_1$ | 6<br>15<br>15<br>43 | 6<br>15<br>20<br>56 | 6<br>15<br>36<br>99 |
| | 6 | 6 | 6 |

| | | | |
|---|---|---|---|
| $X_2$ | 20<br>15<br>56 | 20<br>20<br>76 | 20<br>36<br>131 |
| $X_3$ | 6<br>36<br>15<br>99 | 6<br>36<br>20<br>131 | 6<br>36<br>36<br>232 |

Table 12

***Step 3:*** Calculate the covariance matrix from knowledge entity 4 using the following equation. If $i = j$ the covariance is the variance. Each of the terms used in the covariance matrix are available from the composite knowledge entity shown in Table 12.

| | $X_J$ |
|---|---|
| $X_i$ | $Covar_{ij} = \frac{\Sigma X_i X_j - \frac{(\Sigma X_i \Sigma X_j)}{N_{ij}}}{N_{ij}}$ |

Table 13

The resulting covariance matrix from Table 12 is set out below at Table 14.

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $X_1$ | 0.916666667 | 1 | 1.5 |
| $X_2$ | 1 | 1.555555556 | 1.833333333 |
| $X_3$ | 1.5 | 1.833333333 | 2.666666667 |

Table 14

***Step 4:*** Calculate the correlation matrix from the covariance matrix using the following equation.

| | $X_J$ |
|---|---|
| $X_i$ | $R_{ij} = \frac{Covar_{ij}}{\sqrt{Var_i \, Var_j}}$<br>where:<br>$Var_i = Covar_{ii}$<br>$Var_j = Covar_{jj}$ |

Table 15

Correlation matrix:

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $X_1$ | 1 | 0.837435789 | 0.959403224 |
| $X_2$ | 0.837435789 | 1 | 0.900148797 |
| $X_3$ | 0.959403224 | 0.900148797 | 1 |

Table 16

***Step 5:*** Select the dependent variable $y$ ($X_3$) and then slice the correlation matrix to a matrix for the independent variables $\boldsymbol{R}_{ij}$ and a vector for the dependent variable $\boldsymbol{R}_{yj}$. Calculate the population coefficient $\beta_j$ for independent variables $X_j$ using the relationship.

$$\square_j = \boldsymbol{R}^{-1}{}_{ij}\,\boldsymbol{R}_{yj}$$

From Table 16, a dependent variable correlation vector $R_{yj}$ is obtained as shown in Table 17.

| $X_3$ |
|---|
| 0.959403224 |
| 0.900148797 |

Table 17

Similarly, the independent variables correlation matrix $R_{ij}$ and its inverse matrix $R_{ij}^{-1}$ for $X_1$ and $X_2$ is obtained from Table 16 as set forth below at Tables 18 and 19 respectively.

| | $X_1$ | $X_2$ |
|---|---|---|
| $X_1$ | 1 | 0.837435789 |
| $X_2$ | 0.837435789 | 1 |

Table 18

| | $X_1$ | $X_2$ |
|---|---|---|
| $X_1$ | 3.347826087 | -2.803589382 |
| $X_2$ | -2.803589382 | 3.347826087 |

Table 19

Calculate □ vector for Table 17 and 19 to obtain:

| □ |
|---|
| 0.68826753 |
| 0.32376893 |

Table 20

***Step 6:*** Calculate sample coefficients $b_j$

$b_j = \square_j(s_y / s_j)$

$s_y$ is the sample standard deviation of dependent variable $\boldsymbol{X_3}$ and $s_j$ the sample standard deviation of independent variables ($\boldsymbol{X_1}$, $\boldsymbol{X_2}$)which can be easily calculated from the knowledge entity 46.

$b_1 = 0.68826753 * (1.788854382 * 1.048808848) = 1.173913043 = 1.174$

$b_2 = 0.32376893 * (1.788854382 * 1.366260102) = 0.423913043 = 0.424$

***Step*** *7:* Calculate intercept $a$ from the following equation (Y is $\boldsymbol{X_3}$ in our example):

$$a = \overline{Y} - b_1\overline{X_1} - b_2\overline{X_2} - \ldots - b_n\overline{X_n}$$

where any mean value can be calculated from $\square\ X_i / N_{ii}$

$a = 6 - (1.174 * 2.5) - (0.424 * 3.3333) = 1.652173913 = 1.652$

***Step 8:*** Finally the linear equation which can be used for the prediction.

$\boldsymbol{X_3} = 1.652 + 1.174 * \boldsymbol{X_1} + 0.424 * \boldsymbol{X_2}$

which will be recognised as the same equation calculated from whole dataset.

The above examples have used a linear regression model. Using the knowledge entity 46, the analytical engine can also develop intelligent versions of other models, including, but not limited to, non-linear regression, linear classification, non-linear classification, robust Bayesian classification, naïve Bayesian classification, Markov chains, hidden Markov models, principal component analysis, principal component regression, partial least squares, and decision tree.

An example of each of these will be provided, utilising the data obtained from the process of Figure 1. Again, it will be recognised that this procedure is not process dependent but may be used with any set of data.

## LINEAR CLASSIFICATION

As mentioned above, effective scenario testing depends upon being able to examine a wide variety of mathematical models to see future possibilities and assess relationships amongst variables while examining how well the existing data is explained and how well new results can be predicted. The analytical engine enables provides an extremely effective method for accomplishing scenario testing. One important attribute is that it enables many different modeling methods to be examined including some that involve qualitative (categorical) as well as quantitative (numerical) quantities. Classification is used when the output (dependent) variable is a categorical variable. Categorical variables can take on distinct values, such as colours (red, green, blue) or sizes (small, medium, large). In the embodiment of the dryer 10, a filter may be provided in the vent 20, and optionally removed. A categorical variable for the filter has possible values "on" and "off" reflective of the status of the filter. Suppose the dependent variable $X_i$ has k values. Instead of just one regression model we build k models by using the same steps as set out above with reference to a model using linear regression .

$$X_{i1} = a_1 + b_{11}X_1 + b_{21}X_2 + \ldots + b_{n1}X_n$$

$$X_{i2} = a_2 + b_{12}X_1 + b_{22}X_2 + \ldots + b_{n2}X_n$$

$$\ldots$$

$$X_{ik} = a_k + b_{1k}X_1 + b_{2k}X_2 + \ldots + b_{nk}X_n$$

In the prediction phase, each of the models for $X_{i1}, \ldots, X_{ik}$ is used to construct an estimate corresponding to each of the k possible values. The $k$ models compete with each other and the model with the highest value will be the winner, and determines the predicted one of the k possible values. Using the following equation will transform the actual value to probability.

$$P\,(X_{ik}) = 1\,/\,(1 + \exp\,(-X_{ik}))$$

Suppose we have a model with two variables $(X_{1,}\ X_2)$ and $X_2$ is a categorical variable with values (A, B). In the example of the dryer, A corresponds to the filter being on, and B corresponds to the filter being off. The knowledge entity 46 for this model is going to have one column/row for any categorical value $(X_{2A}, X_{2B})$

$$X_{2A} = a_A + b_{1B}X_1$$

$$X_{2B} = a_B + b_{1B}X_1$$

Table 21 shows a knowledge entity 46 with a categorical variable $X_2$.

| | | $X_1$ | $X_2$ | |
|---|---|---|---|---|
| | | $X_1$ | $X_{2A}$ | $X_{2B}$ |
| $X_1$ | $X_1$ | $N_{11}$<br>□ $X_1$<br>□ $X_1$<br>□ $X_1\,X_1$ | $N_{12A}$<br>□ $X_1$<br>□ $X_{2A}$<br>□ $X_1\,X_{2A}$ | $N_{12B}$<br>□ $X_1$<br>□ $X_{2B}$<br>□ $X_1\,X_{2B}$ |

| | | | | |
|---|---|---|---|---|
| $X_2$ | $X_{2A}$ | $\boldsymbol{N_{2A1}}$<br>□ $X_{2A}$<br>□ $X_1$<br>□ $X_{2A}$ $X_1$ | $\boldsymbol{N_{2A2A}}$<br>□ $X_{2A}$<br>□ $X_{2A}$<br>□ $X_{2A}$ $X_{2A}$ | $\boldsymbol{N_{2A2B}}$<br>□ $X_{2A}$<br>□ $X_{2B}$<br>□ $X_{2A}$ $X_{2B}$ |
| | $X_{2B}$ | $\boldsymbol{N_{2B1}}$<br>□ $X_{2B}$<br>□ $X_1$<br>□ $X_{2B}$ $X_1$ | $\boldsymbol{N_{2B2A}}$<br>□ $X_{2B}$<br>□ $X_{2A}$<br>□ $X_{2B}$ $X_{2A}$ | $\boldsymbol{N_{2B2B}}$<br>□ $X_{2B}$<br>□ $X_{2B}$<br>□ $X_{2B}$ $X_{2B}$ |

Table 21

Table 22 shows a knowledge entity 46 for $X_{2A}$

| | | | |
|---|---|---|---|
| | | $X_1$ | $X_2$ |
| | | $X_1$ | $X_{2A}$ |
| $X_1$ | $X_1$ | $\boldsymbol{N_{11}}$<br>□ $X_1$<br>□ $X_1$<br>□ $X_1$ $X_1$ | $\boldsymbol{N_{12A}}$<br>□ $X_1$<br>□ $X_{2A}$<br>□ $X_1$ $X_{2A}$ |

| | | | |
|---|---|---|---|
| $X_2$ | $X_{2A}$ | $\boldsymbol{N}_{2A1}$<br>□ $X_{2A}$<br>□ $X_1$<br>□ $X_{2A}$ $X_1$ | $\boldsymbol{N}_{2A2A}$<br>□ $X_{2A}$<br>□ $X_{2A}$<br>□ $X_{2A}$ $X_{2A}$ |

Table 22

Table 23 shows a knowledge entity 46 for $X_{2B}$

| | | $X_1$ | $X_2$ |
|---|---|---|---|
| | | $X_1$ | $X_{2B}$ |
| $X_1$ | $X_1$ | $\boldsymbol{N}_{11}$<br>□ $X_1$<br>□ $X_1$<br>□ $X_1$ $X_1$ | $\boldsymbol{N}_{12B}$<br>□ $X_1$<br>□ $X_{2B}$<br>□ $X_1$ $X_{2B}$ |
| $X_2$ | $X_{2B}$ | $\boldsymbol{N}_{2B1}$<br>□ $X_{2B}$<br>□ $X_1$<br>□ $X_{2B}$ $X_1$ | $\boldsymbol{N}_{2B2B}$<br>□ $X_{2B}$<br>□ $X_{2B}$<br>□ $X_{2B}$ $X_{2B}$ |

Table 23

The knowledge entity 46 shown in Tables 22 and 23 may then be applied to model each value of the categorical variable $X_2$. Prediction of the categorical variable is then performed by predicting a score for each possible value. The possible value with the highest score is chosen as the value of the categorical variable. The analytical engine thus enables the development of models which involve categorical as well as numerical variables

## NON-LINEAR REGRESSION AND CLASSIFICATION

The analytical engine is not limited to the generation of linear mathematical models. If the appropriate model is non-linear, then the knowledge entity shown in Figure 3 is also used. The combinations used in the table are sufficient to compute the non-linear regression.

The method of Figure 7 showed how to expand the knowledge entity 46 to include additional variables. This feature also allows the construction of non-linear regression or classification models. It is noted that non-linearity is about variables not coefficients. Suppose we have a linear model with two variables ($X_1$, $X_2$) but we believe $Log\ (X_1)$ could give us a better result. The only thing we need to do is to follow the three steps for adding a new variable. $Log\ (X_1)$ will be the third variable in the knowledge entity 46 and a regression model can be constructed in the explained steps. If we do not need $X_1$ anymore it can be removed by using the contraction feature described above.

| | $X_1$ | $X_2$ | $X_3 = Log\ (X_1)$ |
|---|---|---|---|
| $X_1$ | $N_{11}$<br>□ $X_1$<br>□ $X_1$ | $N_{12}$<br>□ $X_1$<br>□ $X_2$ | $N_{13}$<br>□ $X_1$<br>□ $X_3$ |

| | | | |
|---|---|---|---|
| | □ $X_1 X_1$ | □ $X_1 X_2$ | □ $X_1 X_3$ |
| $X_2$ | $N_{21}$<br>□ $X_2$<br>□ $X_1$<br>□ $X_2 X_1$ | $N_{22}$<br>□ $X_2$<br>□ $X_2$<br>□ $X_2 X_2$ | $N_{23}$<br>□ $X_2$<br>□ $X_3$<br>□ $X_2 X_3$ |
| $X_3$ | $N_{31}$<br>□ $X_3$<br>□ $X_1$<br>□ $X_3 X_1$ | $N_{32}$<br>□ $X_3$<br>□ $X_2$<br>□ $X_3 X_2$ | $N_{33}$<br>□ $X_3$<br>□ $X_3$<br>□ $X_3 X_3$ |

Table 24

Once the knowledge entity 46 has been constructed, the learner 44 can acquire data as shown in Figure 7. The new variable $X_3$ notionally represents a new sensor which measures the logarithm of $X_1$. However, values of the new variable $X_3$ may be computed from values of $X_1$ by a processor rather than by a special sensor. Regardless of how the values are obtained, the learner 44 builds the knowledge entity 46. Then the modeller 48 determines a linear regression of the three variables $X_1$, $X_2$, $X_3$, where $X_3$ is a non-linear function of $X_1$. It will therefore be recognised that operation of the controller 40 is similar for the non-linear regression when the variables are regarded as $X_1$, $X_2$, and $X_3$. The predictor 50 can use a model such as $X_2 = a + b_1 X_1 + b_3 X_3$ to predict variables such as $X_2$.

## DIMENSION REDUCTION

As stated earlier, reducing the number of variables in a model is termed "dimension reduction". Dimension reduction can be done by deleting a variable. As shown earlier, using the knowledge entity the analytical engine easily accommodates this without using the whole database and a tedious re-calibration or re-training step. Such dimension reduction can also be done by the analytical engine using the sum of two variables or the difference between two variables as a new variable. Again, the knowledge entity permits this step to be done expeditiously and makes extremely comprehensive testing of different combinations of variable practical, even with very large data sets. Suppose we have a knowledge entity with three variables but we want to decrease the dimension by adding two variables ($X_1$, $X_2$). For example, the knowledge elements in the knowledge entity associated with the new variable $X_4$ which is the sum of two other variables, $X_1$ and $X_2$ are calculated as follows:

$$
\begin{aligned}
&(1)\ X_4 = X_1 + X_2 \\
&(2)\ \sum X_4 = \sum (X_1 + X_2) \\
&\qquad = \sum X_1 + \sum X_2 \\
&(3)\ \sum X_4 X_3 = \sum (X_1 + X_2) X_3 \\
&\qquad = \sum X_1 X_3 + \sum X_2 X_3 \\
&(4)\ \sum X_4 X_4 = \sum (X_1 + X_2)(X_1 + X_2) \\
&\qquad = \sum X_1 X_1 + 2\sum X_1 X_2 + \sum X_2 X_2
\end{aligned}
$$

Table 25

This is a recursive process and can decrease a model with N dimensions to just to one dimension if it is needed. That is, a new variable $X_5$ can be defined as the sum of $X_4$ and $X_3$.

Alternatively, if we decide to accomplish the dimension reduction by subtracting the two variables, then the relevant knowledge elements for the new variable $X_4$ are:

$$
\begin{aligned}
&(1)\ X_4 = X_1 - X_2\\
&(2)\ \sum X_4 = \sum (X_1 - X_2)\\
&\qquad = \sum X_1 - \sum X_2\\
&(3)\ \sum X_4 X_3 = \sum (X_1 - X_2) X_3\\
&\qquad = \sum X_1 X_3 - \sum X_2 X_3\\
&(4)\ \sum X_4 X_4 = \sum (X_1 - X_2)(X_1 - X_2)\\
&\qquad = \sum X_1 X_1 - 2\sum X_1 X_2 + \sum X_2 X_2
\end{aligned}
$$

Table 26

The knowledge elements in the above tables can all be obtained from the knowledge elements in the original knowledge entity obtained from the original data set. That is, the knowledge entity computed for the models without dimension reduction provides the information needed for construction of the knowledge entity of the dimension reduced models.

Now, returning to the example of Table 4 showing the output rates for three different dryers the knowledge entity for the sample dataset is:

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $X_1$ | $N_{11} = 6$<br>□ $X_1$ =15<br>□ $X_1$ =15<br>□ $X_1X_1$ =43 | $N_{12} = 6$<br>□ $X_1$ =15<br>□ $X_2$ =20<br>□ $X_1X_2$ =56 | $N_{13} = 6$<br>□ $X_1$ =15<br>□ $X_3$ =36<br>□ $X_1X_3$ =99 |
| $X_2$ | $N_{21} = 6$<br>□ $X_2$ =20<br>□ $X_1$ =15<br>□ $X_2X_1$ =56 | $N_{22} = 6$<br>□ $X_2$ =20<br>□ $X_1$ =20<br>□ $X_2X_2$ =76 | $N_{23} = 6$<br>□ $X_2$ =20<br>□ $X_3$ =36<br>□ $X_2X_3$ =131 |
| $X_3$ | $N_{31} = 6$<br>□ $X_3$ =36<br>□ $X_1$ =15<br>□ $X_3X_1$ =99 | $N_{32} = 6$<br>□ $X_3$ =36<br>□ $X_2$ =20<br>□ $X_3X_2$ =131 | $N_{33} = 6$<br>□ $X_3$ =36<br>□ $X_3$ =36<br>□ $X_3X_3$ =232 |

Table 27

Table 27 has the same quantities as did Table 12. Table12 was calculated by combining the knowledge entities from data obtained from dividing the original data set into three portions (to illustrate distributed processing and parallel processing). The above knowledge entity was calculated from the original undivided dataset.

Now, to show dimension reduction can be accomplished by means other than removal of a variable, the data set for variables $X_4$ and $X_3$ (where $X_4$=$X_1$+$X_2$) is:

| $X_4$=$X_1$+$X_2$ | $X_3$ |
|---|---|
| 5 | 5 |
| 7 | 7 |
| 2 | 3 |
| 5 | 6 |
| 8 | 8 |
| 8 | 7 |

Table 28

The knowledge entity for the $X_4$, $X_3$ data set above is:

|  | $X_4$ | $X_3$ |
|---|---|---|
| $X_4$ | $N_{44}$ = 6<br>□ $X_4$ =35<br>□ $X_4$ =35<br>□ $X_4X_4$ =231 | $N_{43}$ = 6<br>□ $X_4$ =35<br>□ $X_3$ =36<br>□ $X_4X_3$ =230 |
|  | $N_{34}$ = 6 | $N_{33}$ = 6 |

| | | |
|---|---|---|
| $X_3$ | □ $X_3$ =36<br>□ $X_4$ =35<br>□ $X_3 X_4$ =230 | □ $X_3$ =36<br>□ $X_3$ =36<br>□ $X_3 X_3$ =232 |

Table 29

Note that exactly the same knowledge entity can be obtained from the knowledge entity for all three variables and the use of the expressions in Table 25 above.

| | $X_4$ | $X_3$ |
|---|---|---|
| $X_4$ | $N_{44} = 6$<br>□ $X_4$ =*15+20* =35<br>□ $X_4$ =*15+20* =35<br>□ $X_4 X_4$ =*43+(2*56)+76* =231 | $N_{43} = 6$<br>□ $X_4$ =*15+20* =35<br>□ $X_3$ =36<br>□ $X_4 X_3$ =*99+131*=230 |
| $X_3$ | $N_{34} = 6$<br>□ $X_3$ =36<br>□ $X_4$ =*15+20* =35<br>□ $X_3 X_4$ =*99+131*=230 | $N_{33} = 6$<br>□ $X_3$ =36<br>□ $X_3$ =36<br>□ $X_3 X_3$ =232 |

Table 30

## DYNAMIC QUERIES

The analytical engine can also enable “dynamic queries” to select one or more sequences of a series of questions based on answers given to the questions so as to rapidly converge on one or more outcomes. The Analytical Engine can be used with different models to derive the “next best question” in the dynamic query. Two of the most important are regression models and classification models. For example, regression models can be used by obtaining the correlation matrix from the knowledge entity

The Correlation Matrix:

Then, the following steps are carried out:

*Step 1:* Calculate the covariance matrix. (Note: if $i = j$ the covariance is the variance.)

| | $X_1$ | ... | $X_j$ | ... | $X_n$ |
|---|---|---|---|---|---|
| $X_1$ | $r_{11}$ | ... | $r_{1j}$ | ... | $r_{1n}$ |
| ... | ... | ... | ... | ... | ... |
| $X_i$ | $r_{i1}$ | ... | $r_{ij}$ | ... | $r_{in}$ |
| ... | ... | ... | ... | ... | ... |
| $X_m$ | $r_{m1}$ | ... | $r_{mj}$ | ... | $r_{mn}$ |

Table 31

| | $X_J$ |
|---|---|
| $X_i$ | $Covar_{ij} = \dfrac{\sum X_i X_j - \dfrac{\sum X_i \sum X_j}{N_{ij}}}{N_{ij}}$ |

Table 32

*Step 2:* Calculate the correlation matrix from the covariance matrix. (Note: if $i = j$ the elements of the matrix are unity.)

| | $X_J$ |
|---|---|
| | |

| | |
|---|---|
| $X_i$ | $r_{ij} = \dfrac{Covar_{ij}}{\sqrt{Var_i \times Var_j}}$<br>where:<br>$Var_i = Covar_{ii}$<br>$Var_j = Covar_{jj}$ |
| | |

Table 33

Once these steps are completed the Analytical Engine can supply the "next best question" in a dynamic query as follows:

1. Select the dependent variable $X_d$.

2. Select an independent $X_i$ with the highest correlation to $X_d$. If $X_i$ has already been selected, select the next best one.

3. Continue till there is no independent variables or some criteria has been met (e.g., no significance change in R2).

Classification methods can also be used by the Analytical Engine to supply the next best question. The analytical engine selects the variable to be examined next (the "next best question") in order to obtain the maximum impact on the target probability (e.g. probability of default in credit assessment). The user can decide at what point to stop asking questions by examining that probability.

The general structure of this Knowledge Entity for using classification for dynamic query is

| | $X_1$ | ... | $X_j$ | ... | $X_n$ |
|---|---|---|---|---|---|
| $X_1$ | $N_{11}$ | ... | $N_{1j}$ | ... | $N_{1n}$ |
| ... | ... | ... | ... | ... | ... |
| $X_i$ | $N_{i1}$ | ... | $N_{ij}$ | ... | $N_{in}$ |
| ... | ... | ... | ... | ... | ... |
| $X_m$ | $N_{m1}$ | ... | $N_{mj}$ | ... | $N_{mn}$ |

Table 34

where the **...** are “ditto” marks.

The analytical engine uses this knowledge entity as follows:

1. Calculate $T_j = \square\ N_{ij}$ $(i=1...m\ ; j=1...n)$
2. Select $X_c$ (column variables, $c=1...n$) with the highest $T$. If $X_c$ has already been selected, select the next best one.
3. Calculate $S_i = S_i$ x $(N_{ic} / N_{ii})$ or $S_i = S_i$ x $(N_{ic} / \square\ N_{ic})$ for all variables $(i=1....m)$
4. Select $X_r$ (row variables, $r=1...m$) with the highest $S$. If $X_r$ has already been selected, select the next best one.
5. Select Rule Out (Exclude) or Rule In (Include) strategy
    a. Rule Out: calculate $T_j = N_{rj} / N_{rr}$ for all variables where $X_r <> X_j$ $(j=1...n)$

b. Rule In: calculate $T_j = N_{rj} / \square\ N_{ij}$ for all variables where $X_r <> X_j$ $(j=1 \ldots n\ ;\ i=1 \ldots m)$

6. Go to step 2 and repeat steps 2 through 5 until the desired target probability is reached or exceeded.

## NORMALIZED KNOWLEDGE ENTITY

Some embodiments preferably employ particular forms of the knowledge entity. For example, if the knowledge elements are normalized the performance of some modeling methods can be improved. A normalized knowledge entity can be expressed in terms of well known statistical quantities termed "Z" values. To do this,

$\square\ X_i$, $\square\ X_i X_j$, □and □can be extracted from the un-normalized knowledge entity and used as shown below: Then, returning again to the three dryer data of Table 4

$$\text{(1)}\quad \mathrm{Z_i} = \frac{\mathrm{X_i} - \mu_i}{\sigma_i}$$

$$\text{(2)}\quad \sum Z_i = \sum \frac{X_i - \mu_i}{\sigma_i} = \frac{\sum X_i - N\mu_i}{\sigma_i}$$

$$= \frac{\sum X_i - \sum X_i}{\sigma_i} = 0$$

$$\text{(3)}\quad \sum Z_i Z_j = \sum \left( \frac{X_i - \mu_i}{\sigma_i} \times \frac{X_j - \mu_j}{\sigma_j} \right)$$

$$= \sum \left( \frac{X_i X_j - X_i \mu_j - \mu_i X_j + \mu_i \mu_j}{\sigma_i \sigma_j} \right)$$

$$= \frac{\sum X_i X_j - \mu_j \sum X_i - \mu_i \sum X_j + \left( \frac{n_i + n_j}{2} \right) \mu_i \mu_j}{\sigma_i \sigma_j}$$

*where*:

$$\mu_i = \frac{\sum X_i}{N_i} \quad , \quad \mu_j = \frac{\sum X_j}{N_j}$$

$$\sigma_i = \sqrt{\frac{\sum X_i X_i - \frac{\sum X_i}{N_i}}{N_i}} \quad , \quad \sigma_j = \sqrt{\frac{\sum X_j X_j - \frac{\sum X_j}{N_j}}{N_j}}$$

Table 35

The un-normalized knowledge entity was given in Table 12. and the normalized one is provided below.

## NORMALIZED KNOWLEDGE ENTITY FOR THE SAMPLE DATASET:

| | $Z_1$ | $Z_2$ | $Z_3$ |
|---|---|---|---|
| $Z_1$ | $N_{11} = 6$<br>□ $Z_1$ =0<br>□ $Z_1$ =0<br>□ $Z_1Z_1$ =6 | $N_{12} = 6$<br>□ $Z_1$ =0<br>□ $Z_2$ =0<br>□ $Z_1Z_2$ =5.024615 | $N_{13} = 6$<br>□ $Z_1$ =0<br>□ $Z_3$ =0<br>□ $Z_1Z_3$ =5.756419 |
| $Z_2$ | $N_{21} = 6$<br>□ $Z_2$ =0<br>□ $Z_1$ =0<br>□ $Z_2Z_1$ =5.024615 | $N_{22} = 6$<br>□ $Z_2$ =0<br>□ $Z_1$ =0<br>□ $Z_2Z_2$ =6 | $N_{23} = 6$<br>□ $Z_2$ =0<br>□ $Z_3$ =0<br>□ $Z_2Z_3$ =5.400893 |
| $Z_3$ | $N_{31} = 6$<br>□ $Z_3$ =0<br>□ $Z_1$ =0<br>□ $Z_3Z_1$ =5.756419 | $N_{32} = 6$<br>□ $Z_3$ =0<br>□ $Z_2$ =0<br>□ $Z_3Z_2$ =5.400893 | $N_{33} = 6$<br>□ $Z_3$ =0<br>□ $Z_3$ =0<br>□ $Z_3Z_3$ =6 |

Table 36

## SERIALIZED KNOWLEDGE ENTITY

It is also possible to serialize and disperse the knowledge entity to facilitate some software applications.

The general structure of the knowledge entity:

| | $X_1$ | ... | $X_j$ | ... | $X_n$ |
|---|---|---|---|---|---|
| $X_1$ | $W_{11}$ | ... | $W_{1j}$ | ... | $W_{1n}$ |
| ... | ... | ... | ... | ... | ... |

| | | | | | |
|---|---|---|---|---|---|
| $\boldsymbol{X_1}$ | $W_{i1}$ | … | $W_{ij}$ | … | $W_{in}$ |
| ••• | … | … | … | … | … |
| $\boldsymbol{X_m}$ | $W_{m1}$ | … | $W_{mj}$ | … | $W_{mn}$ |
| | | | | | |

Table 37

can be written as the serialized and dispersed structure:

| | | |
|---|---|---|
| $\boldsymbol{X_1}$ | $\boldsymbol{X_1}$ | $W_{11}$ |
| $\boldsymbol{X_1}$ | $\boldsymbol{X_j}$ | $W_{1j}$ |
| $\boldsymbol{X_1}$ | $\boldsymbol{X_n}$ | $W_{1n}$ |
| ••• | ••• | … |
| $\boldsymbol{X_i}$ | $\boldsymbol{X_1}$ | $W_{i1}$ |
| $\boldsymbol{X_i}$ | $\boldsymbol{X_j}$ | $W_{ij}$ |
| $\boldsymbol{X_i}$ | $\boldsymbol{X_n}$ | $W_{in}$ |
| ••• | ••• | … |
| $\boldsymbol{X_m}$ | $\boldsymbol{X_1}$ | $W_{m1}$ |
| $\boldsymbol{X_m}$ | $\boldsymbol{X_j}$ | $W_{mj}$ |
| $\boldsymbol{X_m}$ | $\boldsymbol{X_n}$ | $W_{mn}$ |

Table 38

then the knowledge entity for the three dryer data (Table 4) used above becomes:

| | | | | | |
|---|---|---|---|---|---|
| $\boldsymbol{X_1}$ | $\boldsymbol{X_1}$ | $N_{11} = 6$ | □ $X_1$ =15 | □ $X_1$ =15 | □ $X_1X_1$ =43 |
| $\boldsymbol{X_1}$ | $\boldsymbol{X_2}$ | $N_{12} = 6$ | □ $X_1$ =15 | □ $X_2$ =20 | □ $X_1X_2$ =56 |

| | | | | | |
|---|---|---|---|---|---|
| $\boldsymbol{X_1}$ | $\boldsymbol{X_3}$ | $N_{13}=6$ | □ $X_1$ =15 | □ $X_3$ =36 | □ $X_1X_3$ =99 |
| $\boldsymbol{X_2}$ | $\boldsymbol{X_2}$ | $N_{22}=6$ | □ $X_2$ =20 | □ $X_2$ =20 | □ $X_2X_2$ =76 |
| $\boldsymbol{X_2}$ | $\boldsymbol{X_3}$ | $N_{23}=6$ | □ $X_2$ =20 | □ $X_3$ =36 | □ $X_2X_3$ =131 |
| $\boldsymbol{X_3}$ | $\boldsymbol{X_3}$ | $N_{33}=6$ | □ $X_3$ =36 | □ $X_3$ =36 | □ $X_3X_3$ =232 |

Table 39

## ROBUST BAYESIAN CLASSIFICATION

In some cases, the appropriate model for classification of a categorical variable may be Robust Bayesian Classification, which is based on *Bayes's rule* of conditional probability:

$$P(C_k\,|\mathbf{x}) = \frac{P(\mathbf{x}|C_k)\,P(C_k)}{P(\mathbf{x})}$$

Where:

$P(C_k\,|\mathbf{x})$ is the conditional probability of $C_k$ given **x**

$P(\mathbf{x}|C_k)$ is the conditional probability of **x** given $C_k$

$P(C_k)$ is the prior probability of $C_k$

$P(\mathbf{x})$ is the prior probability of **x**

Bayes's rule can be summarized in this simple form:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalization factor}}$$

A discriminant function may be based on Bayes's rule for each value k of a categorical variable Y:

$$y_k(\mathbf{x}) = \ln P(\mathbf{x}|C_k) + \ln P(C_k)$$

If each of the class-conditional density functions $P(\mathbf{x}|C_k)$ is taken to be an independent normal distribution, then we have:

$$y_k(\mathbf{x}) = -\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) - \tfrac{1}{2}\ln|\Sigma_k| + \ln P(C_k)$$

There are three elements, which the analytical engine needs to extract from the knowledge entity 46, namely, the mean vector (□ $_k$), the covariance matrix (□$_k$), and the prior probability of $C_k$ ($P(C_k)$).

There are five steps to create the discriminant equation:

*Step1:* Slice out the knowledge entity 46 for any $C_k$ where $C_k$ is a $X_i$ .

*Step2:* Create the □vector by simply using two elements in the knowledge entity 46 □ $X$ and $N$ where □ = □ $X$ /$N$

*Step3:* Create the the covariance matrix (□$_k$), by using four basic elements in the knowledge entity 46 as follows :

$$Covar_{ij} = \frac{\Sigma X_i X_j - \dfrac{(\Sigma X_i \Sigma X_j)}{N_{ij}}}{N_{ij}}$$

*Step4:* Calculate the $P(C_k)$ by using two elements in the knowledge entity 46 □ $X$ and $N$. If $C_k = X_i$ then

$P(X_i) = \Box\, X_i \,/\, N_{ii}$ S

*Step 5 k* discriminant functions

In the prediction phase these *k* models compete with each other and the model with the highest value will be the winner.

## NAÏVE BAYESIAN CLASSIFICATION

It may be desirable to use a simplification of Bayesian Classification when the variables are independent. This simplification is called Naïve Bayesian Classification and also uses *Bayes's rule* of conditional probability:

$$P(C_k\,|\mathbf{x}) = \frac{P(\mathbf{x}|C_k)\,P(C_k)}{P(\mathbf{x})}$$

Where:

$P(C_k\,|\mathbf{x})$ is the conditional probability of $C_k$ given $\mathbf{x}$

$P(\mathbf{x}|C_k)$ is the conditional probability of $\mathbf{x}$ given $C_k$

$P(C_k)$ is the prior probability of $C_k$

$P(\mathbf{x})$ is the prior probability of $\mathbf{x}$

When the variables are independent, Bayes's rule may be written as follows :

$$P(C_k\,|\mathbf{x}) = P(x_1|C_k) \times P(x_2|C_k) \times P(x_3|C_k) \times \ldots \times P(x_n|C_k) \times \frac{P(C_k)}{P(\mathbf{x})}$$

It is noted that $P(\mathbf{x})$ is  a normalization factor.

There are five steps to create the discriminant equation:

Step1: Select a row of the knowledge entity 46 for any $C_k$ and suppose $C_k = X_i$

.

Step2a: If $x_j$ is a value for a categorical variable $X_j$ we have $P(x_j \mid X_i)$ = □ $X_j$ /□ $X_i$. We get □ $X_j$ from $W_{ij}$ and □ $X_i$ from $W_{ii}$.

Step2b: If $x_j$ is a value for a numerical variable $X_j$ we calculate $P(x_j \mid X_i)$ by using a density function like this:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Where:

□ = □ $X_i / N_{ii}$

□$_i$ = $sqrt(Covar_{ii})$

Step3: Calculate the $P(C_k)$ by using two elements in the knowledge entity 46 □ $X$ and $N$. If $C_k = X_i$ then

$P(X_i)$ = □ $X_i / N_{ii}$

Step4: Calculate $P(C_k \mid \mathbf{x})$ using

$$P(C_k|\mathbf{x}) = P(x_1|C_k) \times P(x_2|C_k) \times P(x_3|C_k) \times \dots \times P(x_n|C_k) \times \frac{P(C_k)}{P(\mathbf{x})}$$

In the prediction phase these $k$ models compete with each other and the model with the highest value will be the winner.

## MARKOV CHAIN

Another possible model is a Markov Chain, which is particularly expedient for situations where observed values can be regarded as "states." In a conventional Markov Chain, each successive state depends only on the state immediately before it. The Markov Chain can be used to predict future states.

Let $\boldsymbol{X}$ be a set of states ($X_1, X_2, X_3 \dots X_n$ ) and $\boldsymbol{S}$ be a sequence of random variables ($S_0, S_1, S_2 \dots S_l$) each with sample space $\boldsymbol{X}$. If the probability of transition from state $X_i$ to $X_j$ depends only on state $X_i$ and not to the previous states then the process is said to be a *Markov chain*. A time independent Markov chain is called a *stationary Markov chain*. A stationary Markov chain can be described by an $N$ by $N$ transition matrix, $\boldsymbol{T}$, where $N$ is the state space and with entries $T_{ij}= \mathrm{P}(S_k=X_i \mid S_{k-1}=X_j)$.

In a $k^{th}$ *order Markov chain*, the distribution of $S_k$ depends only on the $k$ variables immediately preceding it. In a 1st order Markov chain, for example, the distribution of $S_k$ depends only on the $S_{k-1}$. The transition matrix $T_{ij}$ for a 1st order Markov chain is the same as $N_{ij}$ in the knowledge entity 46. Table 40 shows the transition matrix $\boldsymbol{T}$ for a 1st order Markov chain extracted from the knowledge entity 46.

| | $X_1$ | … | $X_j$ | … | $X_n$ |
|---|---|---|---|---|---|
| $X_1$ | $N_{11}$ | … | $N_{1j}$ | … | $N_{1n}$ |
| … | … | … | … | … | … |
| $X_i$ | $N_{i1}$ | … | $N_{ij}$ | … | $N_{in}$ |
| … | … | … | … | … | … |
| $X_n$ | $N_{n1}$ | … | $N_{nj}$ | … | $N_{nn}$ |

Table 40

One weakness of a Markov chain is its unidirectionality which means $S_k$ depends just on $S_{k-1}$ not $S_{k+1}$. Using the knowledge entity 46 can solve this problem and even give more flexibility to standard Markov chains. A 1st order Markov chain with a simple graph with two nodes (variables) and a connection as shown in Figure 10.

Suppose $X_1$ and $X_2$ have two states A and B then the knowledge entity 46 will be of the form shown in Table 41.

A

| | | $X_1$ | | $X_2$ | |
|---|---|---|---|---|---|
| | | $X_{1A}$ | $X_{1B}$ | $X_{2A}$ | $X_{2B}$ |
| $X_1$ | $X_{1A}$ | $W_{1A1A}$ | $W_{1A1B}$ | $W_{1A2A}$ | $W_{1A2B}$ |
| | $X_{1B}$ | $W_{1B1A}$ | $W_{1B1B}$ | $W_{1B2A}$ | $W_{1B2B}$ |
| $X_2$ | $X_{2A}$ | $W_{2A1A}$ | $W_{2A1B}$ | $W_{2A2A}$ | $W_{2A2B}$ |
| | $X_{2B}$ | $W_{2B1A}$ | $W_{2B1B}$ | $W_{2B2A}$ | $W_{2B2B}$ |

Table 41

It is noted that $W_{\#A*B}$ indicates the set of combinations of variables at the intersection of row #A and column *B. The use of the knowledge entity 46 produces a bi-directional Markov Chain. It will be recognised that each of the above operations relating to the knowledge entity 46 can be applied to the knowledge entity for the Markov Chain. It is also possible to have a Markov chain with a combination of different order in one knowledge entity 46 and also a continuous Markov chain. These Markov Chains may then be used to predict future states.

## HIDDEN MARKOV MODEL

In a more sophisticated variant of the Markov Model, the states are hidden and are observed through output or evidence nodes. The actual states cannot be directly observed, but the probability of a sequence of states given the output nodes may be obtained.

A Hidden Markov Model (HMM) is a graphical model in the form of a chain. In a typical HMM there is a sequence of state or hidden nodes ***S*** with a set of states ($X_1, X_2, X_3 \ldots X_n$), the output or evidence nodes ***E*** a set of possible outputs ($Y_1$, $Y_2$, $Y_3$ … $Y_n$ ), a transition probability matrix ***A*** for the hidden nodes and a emission probability matrix ***B*** for the output nodes as shown in Figure 11.

Table 42 shows a transition matrix ***A*** for a 1st order Hidden Markov Model extracted from knowledge entity 46.

|  | $X_1$ | … | $X_j$ | … | $X_n$ |
|---|---|---|---|---|---|
| $X_1$ | $N_{11}$ | … | $N_{1j}$ | … | $N_{1n}$ |
| … | … | … | … | … | … |
| $X_i$ | $N_{i1}$ | … | $N_{ij}$ | … | $N_{in}$ |
| … | … | … | … | … | … |
| $X_n$ | $N_{n1}$ | … | $N_{nj}$ | … | $N_{nn}$ |

Table 42

Table 43 shows a transition matrix ***B*** for a 1st order Markov chain extracted from knowledge entity 46

| | $X_1$ | ... | $X_j$ | ... | $X_n$ |
|---|---|---|---|---|---|
| $Y_1$ | $N_{11}$ | ... | $N_{1j}$ | ... | $N_{1n}$ |
| ... | ... | ... | ... | ... | ... |
| $Y_i$ | $N_{i1}$ | ... | $N_{ij}$ | ... | $N_{in}$ |
| ... | ... | ... | ... | ... | ... |
| $Y_n$ | $N_{n1}$ | ... | $N_{nj}$ | ... | $N_{nn}$ |

Table 43

Each of the properties of the knowledge entity 46 can be applied to the standard Hidden Markov Model. In fact we can show a 1st HMM with a simple graph with three nodes (variables) and two connections as shown in Figure 12.

Suppose $X_1$ and $X_2$ have two states (values) A and B and $X_3$ has another two values C and D then the knowledge entity 46 will be as shown in Table 44, which represents a 1st order Hidden Markov Model.

| | | $X_1$ | | $X_2$ | | $X_3$ | |
|---|---|---|---|---|---|---|---|
| | | $X_{1A}$ | $X_{1B}$ | $X_{2A}$ | $X_{2B}$ | $X_{3C}$ | $X_{3D}$ |
| $X_1$ | $X_{1A}$ | $W_{1A1A}$ | $W_{1A1B}$ | $W_{1A2A}$ | $W_{1A2B}$ | $W_{1A3C}$ | $W_{1A3D}$ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | $X_{1B}$ | $W_{1B1A}$ | $W_{1B1B}$ | $W_{1B2A}$ | $W_{1B2B}$ | $W_{1B3C}$ | $W_{1B3D}$ |
| $X_2$ | $X_{2A}$ | $W_{2A1A}$ | $W_{2A1B}$ | $W_{2A2A}$ | $W_{2A2B}$ | $W_{2A3C}$ | $W_{2A3D}$ |
| | $X_{2B}$ | $W_{2B1A}$ | $W_{2B1B}$ | $W_{2B2A}$ | $W_{2B2B}$ | $W_{2B3C}$ | $W_{2B3D}$ |
| $X_3$ | $X_{3C}$ | $W_{3C1A}$ | $W_{3C1B}$ | $W_{3C2A}$ | $W_{3C2B}$ | $W_{3C3C}$ | $W_{3C3D}$ |
| | $X_{3D}$ | $W_{3D1A}$ | $W_{3D1B}$ | $W_{3D2A}$ | $W_{3D2B}$ | $W_{3D3C}$ | $W_{3D3D}$ |

Table 44

The Hidden Markov Model can then be used to predict future states and to determine the probability of a sequence of states given the output and/or observed values.

## PRINCIPAL COMPONENT ANALYSIS

Another commonly used model is Principal Component Analysis (PCA), which is used in certain types of analysis. Principal Component Analysis seeks to determine the most important independent variables.

There are five steps to calculate principal components for a dataset.

*Step1:* Compute the covariance or correlation matrix.

*Step2:* Find its eigenvalues and eigenvectors.

*Step3:* Sort the eigenvalues from large to small.

*Step4:* Name the ordered eigenvalues as $\square_1$, $\square_2$, $\square_\square$ . . . and the corresponding eigenvectors as $v_1$, $v_2$, $v_3$, . .

*Step5:* Select the *k* largest eigenvalues.

The covariance matrix or correlation matrix are the only prerequisites for PCA which are easily can be derived from knowledge entity 46.

The Covariance matrix extracted from knowledge entity 46.

| | $X_J$ |
|---|---|
| $X_i$ | $Covar_{ij} = \dfrac{\Sigma X_i X_j - \dfrac{(\Sigma X_i \Sigma X_j)}{N_{ij}}}{N_{ij}}$ |

Table 45

The Correlation matrix.

| | $X_J$ |
|---|---|
| $X_i$ | $R_{ij} = \dfrac{Covar_{ij}}{\sqrt{Var_i \, Var_j}}$<br>where:<br>$Var_i = Covar_{ii}$ |

| | |
|---|---|
| | $Var_j = Covar_{jj}$ |

Table 46

The principal components may then be used to provide an indication of the relative importance of the independent variables based on the covariance or correlation tables computed from the knowledge entity 46, without requiring re-computation based on the entire collection of data.

It will therefore be recognised that the controller 40 can switch among any of the above models, and the modeller 48 will be able to use the same knowledge entity 46 for the new model. That is, the analytical engine can use the same knowledge entity for many modelling methods. There are many models in addition to the ones mentioned above that can be used by the analytical engine. For example, the OneR Classification Method , Linear Support Vector Machine and Linear Discriminant Analysis are all readily employed by this engine. Pertinent details are provided in the following paragraphs.

**The OneR Method**

The main goal in the OneR Method is to find the best independent (Xj ) variable which can explain the dependent variable (Xi ). If the dependent variable is categorical there are many ways that the analytical engine can find the best dependent variable (e.g. Bayes rule, Entropy, Chi2, and Gini index). All of these ways can employ the knowledge elements of the knowledge entity. If the dependent variable is numerical the correlation matrix (again, extracted from the knowledge entity) can be used by the analytical engine to find the best independent variable. Alternatively, the engine can transform the numerical variable to a categorical variable by a discretization technique.

**Linear Support Vector Machine**

The Linear Support Vector Machine can be modeled by using the covariance matrix. As shown in [0079] the covariance matrix can easily be computed from the knowledge elements of the knowledge entity by the analytical engine.

## Linear Discriminant Analysis

Linear Discriminant Analysis is a classification technique and can be modeled by the analytical engine using the covariance matrix. As shown in [0079] the covariance matrix can easily be computed from the knowledge elements of the knowledge entity.

## Model Diversity

As evident above, use of the analytical engine with even a single knowledge entity can provide extremely rapid model development and great diversity in models. Such easily obtained diversity is highly desirable when seeking the most suitable model for a given purpose. In using the analytical engine, diversity originates both from the intelligent properties awarded to any single model (e.g. addition and removal of variables, dimension reduction) and the property that switching modelling methods does not require new computations on the entire database for a wide variety of modelling methods. Once provided with the models, there are many methods for determining which one is best ("model discrimination") or which prediction is best. The analytical engine makes model generation so comprehensive and easy that for the latter problem, if desired, several models can be tested and the prediction accepted can be the one which the majority of models support.

It will be recognised that certain uses of the knowledge entity 46 by the analytical engine will typically use certain models. The following examples illustrate several areas where the above models can be used. It is noted that the knowledge entity 46 facilitates changing between each of the models for each of the following examples.

The above description of the invention has focused upon control of a process involving numerical values. As will be seen below, the underlying principles are actually much more general in applicability than that.

## CONTROL OF A ROBOTIC ARM

In this embodiment an amputee has been fitted with a robotic arm 200 as shown in Figure 9. The arm has an upper portion 202 and a forearm 204 connected by a joint 205. The movement of the robotic arm depend upon two sensors 206, 208, each of which generate a voltage based upon direction from the person's brain. One of these sensors 208 is termed "Biceps" and is for the upper muscle of the arm. The second 206 is termed "Triceps" and is for the lower muscle. The arm moves in response to these two signals and this movement has one of four possibilities: flexion 210 (the arm flexes), extension 210 (the arm extends), pronation 212 (the arm rotates downwards) and supination 212 (the arm rotates upwards). The usual way of relating movement to the sensor signals would be to gather a large amount of data on what movement corresponds to what sensor signals and to train a classification method with this data. The resulting relationship would then be used without modification to move the arm in response to the signals. The difficulty with this approach is its inflexibity. For example, with wear of parts in the arm the relationship determined from training may no longer be valid and a complete new retraining would be necessary. Other problems can include: the failure of one of the sensors or the need to add a third sensor. The knowledge entity 46 described above may be used by the analytical engine to develop a control of the arm divided into three steps: learner, modeller and predictor. The result is that control of the arm can then adapt to new situations as in the previous example.

The previous example showed a situation where all the variables were numeric and linear regression was used following the learner. This example shows how the learner can employ categorical values and how it can work with a classification method.

Exemplary data collected for use by the robotic arm is as follows:

| ***Biceps*** | ***Triceps*** | ***Movement*** |
|---|---|---|
| 13 | 31 | Flexion |
| 14 | 30 | Flexion |
| 10 | 31 | Flexion |
| 90 | 22 | Extension |
| 87 | 19 | Extension |
| 65 | 15 | Extension |
| 28 | 16 | Pronation |
| 27 | 12 | Pronation |
| 33 | 11 | Pronation |
| 72 | 24 | Supination |
| 70 | 36 | Supination |
| 58 | 28 | Supination |
| … | … | … |

Table 47

The record corresponding to the first measurement of 1: 13, 31, 1, 0, 0, 0 is as follows using the set of combinations $n_{ij}, \sum X_i, \sum X_j, \sum X_i X_j$ is as set out below in Table 48.

| | | | | **Movement** | | | |
|---|---|---|---|---|---|---|---|
| | | **Biceps** | **Triceps** | **Flexion** | **Extension** | **Pronation** | **Supination** |
| | **Biceps** | 1<br>13<br>13<br>169 | 1<br>13<br>31<br>403 | 1<br>13<br>1<br>13 | 1<br>13<br>0<br>0 | 1<br>13<br>0<br>0 | 1<br>13<br>0<br>0 |
| | | 1 | 1 | 1 | 1 | 1 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Triceps** | 31<br>13<br>403 | 31<br>31<br>961 | 31<br>1<br>31 | 31<br>0<br>0 | 31<br>0<br>0 | 31<br>0<br>0 |
| ***Movement*** | **Flexion** | 1<br>1<br>13<br>13 | 1<br>1<br>31<br>31 | 1<br>1<br>1<br>1 | 1<br>1<br>0<br>0 | 1<br>1<br>0<br>0 | 1<br>1<br>0<br>0 |
| | **Extension** | 1<br>0<br>13<br>0 | 1<br>0<br>31<br>0 | 1<br>0<br>1<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 |
| | **Pronation** | 1<br>0<br>13<br>0 | 1<br>0<br>31<br>0 | 1<br>0<br>1<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 |
| | **Supination** | 1<br>0<br>13<br>0 | 1<br>0<br>31<br>0 | 1<br>0<br>1<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 |

Table 48

Once records as shown in Table 48 have been learned by the learner 44 into the knowledge entity 46, the modeller 48 can construct appropriate models of various movements. The predictor can then compute the values of the four models:

Flexion = $a + b_1$ * Biceps + $b_2$ * Triceps

Extension = $a + b_1$ * Biceps + $b_2$ * Triceps

Pronation = $a + b_1$ * Biceps + $b_2$ * Triceps

$$\text{Supination} = a + b_1 * \text{Biceps} + b_2 * \text{Triceps}$$

When signals are received from the Biceps and Triceps sensors the four possible arm movements are calculated. The Movement with the highest value is the one which the arm implements.

## PREDICTION OF THE START CODON IN GENOMES

Each DNA (deoxy-ribonucleic acid) molecule is a long chain of nucleotides of four different types, adenine (A), cytosine (C), thymine (T), and guanine (G). The linear ordering of the nucleotides determines the genetic information. The genome is the totality of DNA stored in chromosomes typical of each species and a gene is a part of DNA sequence which codes for a protein. Genes are expressed by transcription from DNA to mRNA followed by translation from mRNA to protein. mRNA (messenger ribonucleic acid) is chemically similar to DNA, with the exception that the base thymine is replaced with the base uracil (U). A typical gene consists of these functional parts: promoter -> start codon -> exon -> stop codon. The region immediately upstream from the gene is the promoter and there is a separate promoter for each gene. The promoter controls the transcription process in genes and the start codon is a triplet (usually ATG) where the translation starts. The exon is the coding portion of the gene and the start codon is a triplet where the translation stops. Prediction of the start codon from a measured length of DNA sequence may be performed by using the Markov Chain to calculate the probability of the whole sequence. That is, given a sequence ***s***, and given a Markov chain ***M***, the basic question to answer is, "What is the probability that the sequence ***s*** is generated by the Markov chain ***M***? The problems with the conventional Markov chain were described above. Here these problems can cause poor predictability because in fact, in genes the next state, not just the previous state, does affect the structure of the start codon.

**ATTTCTAGGAGTACC...**

| $X_1$ | $X_2$ |
|---|---|
| A | T |
| T | T |
| T | C |
| C | T |
| T | A |
| A | G |
| G | G |
| G | A |
| A | G |
| G | T |
| T | A |
| A | C |
| C | C |
| ... | ... |

Table 49

Classic Markov Chain:

Record 1: A T

| | | $X_1$ | | | |
|---|---|---|---|---|---|
| | | **A** | **C** | **G** | **T** |
| $X_2$ | **A** | 0 | 0 | 0 | 0 |
| | **C** | 0 | 0 | 0 | 0 |
| | **G** | 0 | 0 | 0 | 0 |
| | **T** | 1 | 0 | 0 | 0 |

Table 50

A Markov Chain stored in knowledge entity 46 is constructed as follows:

The first Record 1: 1, 0, 0, 0, 0, 0, 0, 1 is transformed to the table:

| | | $X_1$ | | | | $X_2$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **A** | **C** | **G** | **T** | **A** | **C** | **G** | **T** |
| $X_1$ | **A** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | **C** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **G** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **T** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **A** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$X_2$

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **C** | 1<br>0<br>1<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>1<br>0 |
| | **G** | 1<br>0<br>1<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>0<br>0 | 1<br>0<br>1<br>0 |
| | **T** | 1<br>1<br>1<br>1 | 1<br>1<br>0<br>0 | 1<br>1<br>0<br>0 | 1<br>1<br>0<br>0 | 1<br>1<br>0<br>0 | 1<br>1<br>0<br>0 | 1<br>1<br>0<br>0 | 1<br>1<br>1<br>1 |

Table 51

The knowledge entity 46 is built up by the analytical engine from records relating to each measurements. Controller 40 can then operate to determine the probability that a start codon is generated by the Markov Chain represented in the knowledge entity 46.

## SALES PREDICTION

The next embodiment shows that the model to be used with the learner in the analytical engine can be non-linear in the independent variable. In this embodiment sales from a business are to be related to the number of competitors' stores in the area, average age of the population in the area and the population of the area. The example shows that the presence of a non-linear variable can easily be accommodated by the method. Here, it was decided that the logarithm of the population should be used instead of simply the population. The knowledge entity is then formed as follows:

| No. of Competitors | Average Age | Log (Population) | Sales |
|---|---|---|---|
| 2 | 40 | 4.4 | 850000 |
| 2 | 37 | 4.4 | 1100000 |
| 3 | 36 | 4.3 | 920000 |
| 2 | 31 | 4.2 | 950000 |
| 1 | 42 | 4.6 | 107000 |
| ... | ... | ... | ... |

Table 52

From the record: 2, 40, 4.4, 850000, the knowledge entity 46 is generated as set out below in Table 53.

| | No. of Competitors | Average Age | Log (Population) | Sales |
|---|---|---|---|---|
| **No. of Competitors** | 1<br>2<br>2<br>4 | 1<br>2<br>40<br>80 | 1<br>2<br>4.4<br>8.8 | 1<br>2<br>850000<br>1700000 |
| **Average Age** | 1<br>40<br>2<br>80 | 1<br>40<br>40<br>1600 | 1<br>40<br>4.4<br>176 | 1<br>40<br>850000<br>34000000 |
| **Log (Population)** | 1<br>4.4<br>2<br>8.8 | 1<br>4.4<br>40<br>176 | 1<br>4.4<br>4.4<br>19.36 | 1<br>4.4<br>850000<br>3740000 |
| | 1 | 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| **Sales** | 850000<br>2<br>1700000 | 850000<br>40<br>34000000 | 850000<br>4.4<br>3740000 | 850000<br>850000<br>722500000000 |

Table 53

The sales are modelled using the relationship:

**Sales** $= a + b_1$ * **No. of Competitors** + $b_2$ * **Average Age** + $b_3$ * **Log (Population)**

The coefficients may then be derived from the knowledge entity 46 as described above.

The ability to diagnose the cause of problems, whether in machines or human beings is an important application of the knowledge entity 46.

## DISEASE DIAGNOSIS

In this part we want to use the analytical engine to predict a hemolytic disease of the newborn by means of three variables (sex, blood hemoglobin, and blood bilirubin).

| **Newborn** | **Sex** | **Hemoglobin** | **Bilirubin** |
|---|---|---|---|
| Survival | Female | 18 | 2.2 |
| Survival | Male | 16 | 4.1 |
| Death | Female | 7.5 | 6.7 |
| Death | Male | 3.5 | 4.2 |
| ... | ... | ... | ... |

Table 54

A knowledge entity for constructing a naïve Bayesian classifier would be as follow (just for first and forth records):

Record 1: Survival, Female, 18, 2.2
Record 4: Death, Male, 3.5, 4.2

There is a categorical value then we transform it to numerical one:

Record 1 (transformed): 1, 0, 1, 0, 18, 2.2
Record 4: 0, 1, 0, 1, 3.5, 4.2

|  | Newborn |  | Sex |  |  |  |
|---|---|---|---|---|---|---|
|  | Survival | Death | Female | Male | Hemoglobin | Bilirubin |
| Survival | 2 | 2 | 1 | 1 | 1 | 1 |
|  | 1 | 1 | 1 | 0 | 18 | 2.2 |
|  | 1 | 1 | 1 | 0 | 324 | 4.84 |
| Death | 2 | 2 | 1 | 1 | 1 | 1 |
|  | 1 | 1 | 0 | 1 | 3.5 | 4.2 |
|  | 1 | 1 | 0 | 1 | 12.25 | 17.64 |

Table 55

As we can see this Knowledge entity is not orthogonal and uses three combinations of the variables $(N$, □ $X$ and □ $X^2)$ which are enough to model a naïve Bayesian classifier. The knowledge entity 46 may be used to predict survival or death using the Bayesian classification model described above.

From the above examples, it will be recognised that the knowledge entity of Figure 3 may be applied in many different areas. A sampling of some areas of applicability follows.

## BANKING AND CREDIT SCORING

In banking and credit scoring applications, it is often necessary to determine the risk posed by a client, or other measures of relating to the clients finances. In banking and credit scoring, the following variables are often used.

checking_status, duration, credit_history, purpose, credit_amount, savings_status, employment, installment_commitment, personal_status, other_parties, residence_since, property_magnitude, age, other_payment_plans, housing, existing_credits, job, num_dependents, own_telephone, foreign_worker, credit_assessment. Dynamic query is particularly important in applications such as credit assessment where an applicant is waiting impatiently for a decision and the assessor has many of questions from which to choose. By having the analytical engine select the "next best question" the assessor can rapidly converge on a decision.

## BIOINFORMATICS AND PHARMACEUTICAL SOLUTIONS

The example above showed gene prediction using Markov models. There are many other applications to bioinformatics and pharmaceuticals.

In a microarray, the goal is to find a match between a known sequence and that of a disease.

In drug discovery the goal is to determine the performance of drugs as a function of type of drug, characteristics of patients, etc.

## ECOMMERCE AND CRM

Applications to eCommerce and CRM include email analysis, response and marketing.

### Fraud Detection

In order to detect fraud on credit cards, the knowledge entity 46 would use variables such as number of credit card transactions, value of transactions, location of transaction, etc.

## HEALTH CARE AND HUMAN RESOURCES

To perform diagnosis of the cause of abdominal pain uses approximately 1000 different variables.

In an application to the diagnosis of the presence of heart disease, the variables under consideration are:

> age, sex, chest pain type, resting blood pressure, blood cholesterol, blood glucose, rest ekg, maximum heart rate, exercise induced angina, extent of narrowing of blood vessels in the heart

## PRIVACY AND SECURITY

The areas of privacy and security often require image analysis, finger print analysis, and face analysis. Each of these areas typically involves many variables relating to the image and to attempt to match images and find patterns.

### Retail

In the retail industry, the knowledge entity 46 may be used for inventory control, and
sales prediction.

## SPORTS AND ENTERTAINMENT

The knowledge entity 46 may be used by the analytical engine to collect information on sports events and predict the winner of a future sports event.

The knowledge entity 46 may also be used as a coaching aid.

In computer games, the knowledge entity 46 can manage the data required by the games artificial intelligence systems.

## STOCK AND INVESTMENT ANALYSIS AND PREDICTION

By employing the knowledge entity 46, the analytical engine is particularly adept at handling areas like investment decision making, predicting stock price, where there is a large amount of data which is constantly updated as stock trades are made on the market.

## TELECOM, INSTRUMENTATION AND MACHINERY

The areas of telecom, instrumentation and machinery have many applications, such as diagnosing problems, and controlling robotics.

## TRAVEL

Yet another application of the analytical engine employing the knowledge entity 46 is  as a travel agent. The knowledge entity 46 can collect information about travel preferences, costs of trips, and types of vacations to make predictions related to the particular customer.

From the preceding examples, it will be recognised that the knowledge entity 46 when used with the appropriate methods to form the analytical engine, has broad applicability in many environments. In some embodiments, the knowledge entity 46 has much smaller storage requirements than that required for the equivalent amount of observed data. Some embodiments of the knowledge entity 46 use parallel processing to provide increases in the speed of computations. Some embodiments of the knowledge entity 46 allow models to be changed without re-computation.It will therefore be recognised that in various embodiments, the analytical engine provides an intelligent learning machine that can rapidly learn, predict, control, diagnose, interact, and co-operate in dynamic environments, including for example large quantities of data, and further provides a parallel processing and distributed processing capability.